Auto Labeling to Increase Aspect-Based Sentiment Analysis Using K-Nearest Neighbors Method

Ahmad Jazuli[1], Widowati[2], Retno Kusumaningrum[3]

[1]*Information System, School of Postgraduate Studies, Diponegoro University, Semarang, 50275, Indonesia.*

[2]*Department of Mathematics, Faculty of Sains and Mathematics, Diponegoro University, Semarang, 50275, Indonesia.*

[3]*Department of Informatics, Faculty of Sains and Mathematics, Diponegoro University, Semarang, 50275, Indonesia.*

[a]*ahmadjazuli@students.undip.ac.id*
[b]*widowati@lecturer.undip.ac.id*
[c]*retno@live.undip.ac.id*

Abstract. Social media platforms generate many opinions, emotions, and views on all public services. Sentiment analysis is used in various institutions, such as universities, the business industry, and politicians. The evaluation process requires some data, both quantitative and qualitative. Researchers only focus on quantitative data but ignore qualitative data. The evaluation process given by students in the form of a review is qualitative data that is not structured, so it cannot use conventional methods. Unstructured data requires analysis as well as labeling. The labeling process of large amounts of data is a waste of time and money. Data labeling requires very high accuracy to avoid errors. Accuracy in data labeling is used for the process of classifying, training, and testing data. This study aims to automate data labeling using the K-Nearest Neighbors algorithm method. This labeling process can improve the accuracy of sentiment analysis. The results of the classification method can classify responses from Twitter users and can be used by universities as material for evaluating and assessing higher education services. The results of using a confusion matrix with 1.409 data obtained an accuracy rate of 79.43% with a value of k=15

Keywords. *Auto Labeling, Sentiment Analysis, Learning Process, K-Nearest Neighbors*

1. Introduction

The evaluation process at educational institutions or universities is a process that is carried out dynamically and continuously, the aim is to improve the quality of education [1]. This is done to realize efforts to achieve educational goals effectively and efficiently so as to significant impact on the progress of educational institutions. To carry out the evaluation process, it is necessary to have quantitative and qualitative data [2]. In general, institutions and researchers carry out an educational evaluation process that focuses on quantitative data (Numerical Rating) but ignores qualitative data (student's comments). While in reality qualitative data can be used to obtain more detailed information on the evaluation process [3]. The evaluation process given by students in the form of reviews is qualitative data that is unstructured so that it cannot be analyzed using conventional techniques and a technique called sentiment analysis is needed.

Aspect-based sentiment analysis can dig deep to find out the positive, negative, or neutral reasons for opinions that have been expressed by users [4]. Aspect-based sentiment analysis is generally divided into two sub-tasks, namely Aspect Mining and Aspect Sentiment Classification. Aspect mining sub-tasks can be used to extract words or terms from each review sentence that contains aspects. The supervised model has been applied, the unsupervised model has been applied and the semi-maintained model has been applied to the aspect mining subtask. The aspect sentiment classification sub-task has been widely discussed and can be used to predict sentiment polarity in various aspects. However, many discussions only focus on one sub-task [5]. Consequently, it is necessary to train two different models and use them together for aspect-based sentiment analysis. An unsupervised model has been developed for sentiment analysis of reviews of University performance. Most of the previous studies have applied supervised learning to aspect mining or aspect sentiment classification sub-tasks and require a fair amount of labeled review. However, obtaining labeled data is very expensive and time-consuming, and it is difficult to obtain a lot of labeled data because it requires the efforts of a human expert [6]. So previous researchers were motivated to develop a semi-supervised model for aspect-based sentiment analysis

The following are the contributions of this research:
1. By applying the kernel technique, using discriminant ratios to evaluate class separability in the KNN feature space in data labeling, as well as measuring the level of imbalance in the label subset.
2. Using the KNN algorithm for auto labeling based on active learning to select a subset of labels iteratively for the k-label sets method.
3. Using the KNN algorithm to handle the problem of labeling words or sentences that are not standard and overlapping,

2. Theoretical Background

2.1 Auto Labeling

Supervised learning is a very important topic in research, especially in machine learning. The supervised learning process is built by mapping from feature space to label space [7]. In traditional supervised learning, there are two descriptions of labeling, namely: single-label learning (SLL) and multi-label learning (MLL) [8]. single-label learning anticipates that all training data is labeled with a single label, and multi-label learning allows draining data to be labeled with multiple labels; thus, multi-label learning can discard the more ambiguous case, where one instance is labeled with multiple labels [9], both single-label learning and multi-label learning can answer the basic question of which label can describe the instance, however, neither single-label learning nor multi-label learning can

directly answer further questions with more ability than how many labels were tested. Some data containing information about the relative importance of each label may be more general than we previously thought. Some researchers propose a new paradigm of machine learning to describe supervision as a histogram or probability distribution named Label Distribution Learning (LDL) [10]. Label Distribution Learning has been applied in different real-world scenarios and achieved great success in recent years, such as facial age estimation, indoor Mass Computing, facial expression recognition and text mining in supervised learning applications, the availability of large amounts of labeled data provides opportunities to support performance prediction [11]. It is also possible to identify the relative importance of each possible label when annotating an instance with a label distribution rather than a logical label, leading to higher annotation costs than in traditional multilabel learning.

2.2 Aspect-base Sentiment Analysis

Sentiment analysis aims to evaluate the emotions, opinions, and evaluations conveyed by the speaker or writer towards a product, service, or person's personality. Some research, especially on products, is preceded by determining the discussion of a product before starting the opinion exploration process [12]. Sentiment analysis is the process of understanding and classifying what is contained in writing using text analysis techniques. Sentiment analysis is often referred to as opinion mining. This shows that it will explore what emotions are behind every commentary on the world of education, especially universities. Currently, stakeholders are very happy to express their feelings through platforms such as social media, e-commerce, and websites. One of the advantages of sentiment analysis is that it saves time and effort, as it can be done automatically [13]. Various tools have been equipped with special algorithms to analyze large amounts of data.

2.3 Confusion matrix

The confusion matrix was used as an evaluation of the classification of the K-Nearest Neighbor (KNN) method. By using a confusion matrix, it is expected to know how accurate the method being evaluated is. The parameters used are accuracy, precision, and, recall. Table 2.1 is an example of a confusion matrix table.

Table 2.1. Confusion Matrix

Condition		Condition True Value	
		Classified Positive	Classified Negative
Predicted Value	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Information:
True Positive (TP)   = Number of correct predictions from positive classified data
False Positive (FP)   = Correct Production Amount from negative classified data
True Negative (TN)   = Number of false predictions from true from positive classified data
False Negative (FN)   = Number of wrong predictions from data classified as negative

Accuracy is a calculation to remove the corresponding prediction label from the actual label [14]. If the accuracy results obtained in the calculation are higher than the predictions, then the algorithm used is good. Taking the accuracy value from the result of adding true positive and true negative, the sum of which will be shared with all data amounts can be seen in equation 2.1:

$$P + TN\ accuracy = \frac{T}{TP + FP + TN + FN} \times 100\%$$

Equation 2.1 Accuracy Formula

Precision is a calculation of the comparison of known data in a system with the amount in the document known as the system., it can be seen in equation 2.2:

$$Precision = \frac{TP}{TP + FP}$$

Equation 2.2 Precision Formula

Recall (success) calculation of the total comparison of data related to all total related data, can be seen in equation 2.3*:*

$$Recall = \frac{TP}{TP + FN}$$

Equation 2.3 Recall Formula

The value of the f1-score is a value that is used to determine the performance value of the K-Nearest Neighbor algorithm. The value of the f1-score can be seen in equation 2.4:

$$fl - score = \ 2\ x\ \frac{precision \ast recall}{precision + recall}$$

Equation 2.4. Formula F1-Score

The confusion matrix was used as an evaluation of the classification of the K-Nearest Neighbor method. By using a confusion matrix, it is expected to know how accurate the method being evaluated.

2.3 K-Nearest Neighbor

The K-Nearest Neighbor method is one of the supervised learning methods whose classification process is based on the distance of the closest neighbors from the training dataset [15]. K-Nearest Neighbor is a method that is often used to implement distance calculations in the classification process because it has a simple formula. The advantages of this method are that it has resistance to training data that has a lot of noise and is effectively used for large training data. In addition, the K-NN method also has several weaknesses, such as the unclear type of distance calculation and which should be used to obtain the best results, requires a fairly high computational cost, and the need to determine the value of the k parameter [16]. There are several types of distance calculations, including euclidean, manhattan/city block, cosine, and correlation. The formula for calculating the Euclidean distance can be seen in equation 2.5:

$$d_i = \sqrt{\sum_{i=1}^{n} (X_{2i} - X_{1i})^2}$$

Equation 2.5 Euclidean Distance Formula

Keterangan :
d_i = euclidean distance to-i
X_{2i} = training distance to-i
X_{1i} = testing distance to-i
n = lots of training data
I = line to -i

The K-Nearest Neighbor method is one of the supervised learning methods whose classification process is based on the distance of the closest neighbors from the training dataset. K-Nearest Neighbor is a method that is often used to implement distance calculations in the classification process because it has a simple formula [17]. The advantages of this method are that it has resistance to training data that has a lot of noise and is effectively used for large training data. In addition, the K-NN method also has several weaknesses, such as the unclear type of distance calculation and which should be used to obtain the best results, requires a fairly high computational cost, and the need to determine the value of the k parameter.

3. Research Method
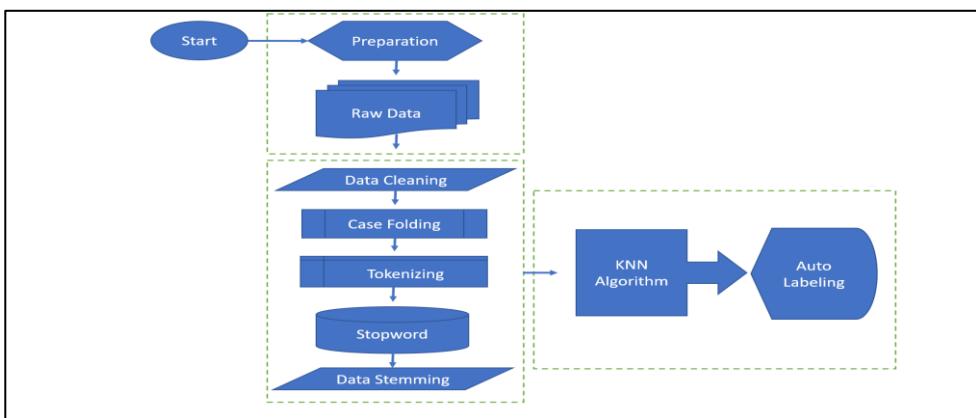
3.1 System Scenario

3.1.1 Crawling

This stage is carried out to retrieve data from Twitter by using the Python programming language and the Twitter API.

3.1.2 Pre-processing

After crawling, the next stage is preprocessing which includes cleansing, case folding, tokenizing, stopword, stemming, and word weighting (labeling). Flowchart or flow chart can be seen in Figure 3.1:
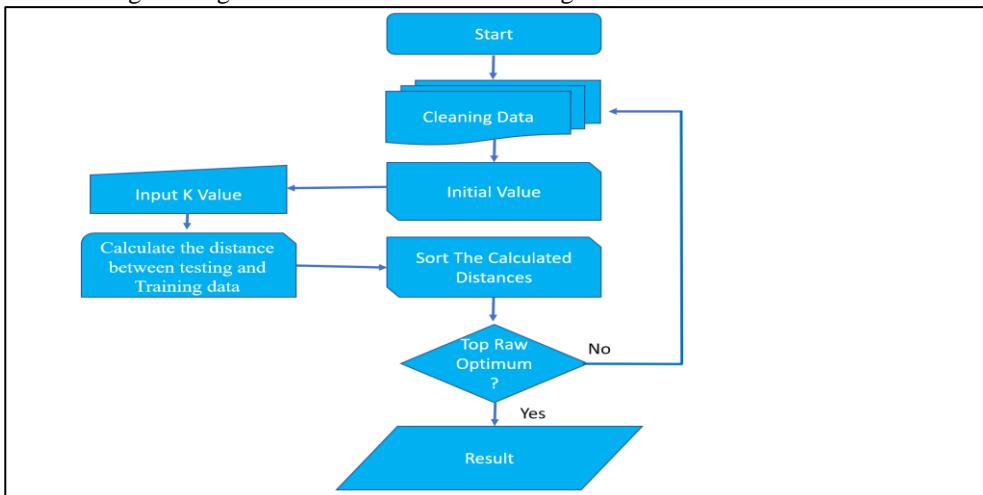

Fig. 3.1 Preprocessing Flowchart

The preprocessing stage begins with the crawled dataset. After that, the case folding stage is the process of changing uppercase letters to lowercase letters [18]. After that is the tokenizing stage, namely the process of separating text words and spaces. Next is the stopword stage, which is the process of eliminating words that are not important. Then the stemming stage is the process of removing affixes to the basic form. Next is the labeling process, labeling can be done from datasets that have been processed. The system is labeled automatically using the textblob library from python to perform natural language processing (NLP). The textblob library will calculate the value of data to get positive, negative, and neutral sentiments. After the dataset is labeled, the data will be classified using the K-Nearest Neighbor method and tested with a confusion matrix [19].

3.1.3 Klasifikasi K-Nearest Neighbor

Classification is done by using the K-Nearest Neighbor algorithm model by using the results of the weighting of each designated word from the preprocessed tweet. The flow of the K-Nearest Neighbor algorithm model can be seen in Figure 3.2:


Fig. 3.2 KNN Classification FlowChart

After the data is preprocessed and then word weighted, the next step is to classify it into the K-Nearest Neighbor method by taking the word weighting results. The net dataset will be divided by a combination of 80% training data and 20% testing data. Determine the value of k, then the training data will be trained by the system to produce the knn model, then the knn model is used to predict the classification results using testing data.

3.2 Evaluation Stage of Calculation Test

At this stage, it is used to determine the accuracy of the K-Nearest Neighbor method by using the Confusion Matrix test. Testing calculations using the Confusion Matrix with equations 2.1, 2.2, 2.3, 2.4. The parameters used are accuracy, precision, recall, and fi-score.

4. Result and Discussion

This research uses the python programming language with a web-based flask framework. The stages of this research process are starting with crawling to collect datasets using the Twitter API and stored with files with the .csv extension. Then the data is processed through the preprocessing process, then the labeling process, and classification using the K-Nearest Neighbor method. The following are the stages in the classification research in this sentiment analysis:

4.1 Dataset Collection

Dataset collection or crawling is done using the Twitter API using the python programming language. The crawling process takes the most recent data and connects it to the internet. To get access to the Twitter API, you must register as a Twitter developer on the website https://developer.twitter.com/en/apply-for-access consisting of an API key, API Key Secret, Access Token, and Access Token Secret. for authentication with the Twitter developer. After connecting to the Twitter API, the next step is crawling with university keywords. The data generated from crawling are in the form of dates, usernames, and tweets with a dataset of 1409 data obtained.

Crawling is done with Visual Studio Code tools with python and flask programming languages. Then the dataset is saved in excel with .csv format.

4.2 Data Preprocessing

After successfully crawling, we can download the dataset file into .csv form, the next process is preprocessing the dataset to clean the data, making sentences that are not standard become standard, and maintaining the consistency of the dataset used. The end of this process can be downloaded in .csv format, while the stages in preprocessing are described as follows:

4.2.1. Cleaning Data

The cleaning stage aims to remove symbols, punctuation marks, numbers, URLs, and mentions and produce sentences that will be processed to the next stage.

4.2.2. Tokenizing

At this stage tokenizing is used to separate word spacing and spaces for data weighting. This tokenizing uses the NLTK library to simplify the tokenizing process by calling the word_tokenize() function. At this stage, normalization is also carried out because the sentence structure and language used are not always by the KBBI. Twitter users tend to use slang, and abbreviations so that the writing is not standard, for example, "don't know" after normalization will become "don't know", the word "lgsg" becomes "langsung", "mkn" becomes "makan" and many more. This can result in the data not being read optimally in conducting sentiment analysis. The thing that needs to be prepared in this process is that researchers must collect as much data as possible to get maximum results. After normalization is done, the data that was not standard becomes standard data.

4.2.3. Stopword

At the stopword stage, unimportant words will be deleted. This stopword process uses the NLTK library by calling the stopwords words ('Indonesia') function and the researcher adds some additional stopwords

4.2.4. Stemming

The stemming process is a process to remove affixes from a tweet so that it becomes a basic word by using a literary library in Indonesian. In this process, the system will track all tweets from beginning to end, and if found words with affixes, then the affixed words will be converted into basic words.

4.2.5. Preprocessing

After doing the cleaning, tokenizing, stopword, and stemming processes. Then obtained the overall results of preprocessing. The data from the preprocessing results can be seen in Figure 3.1 If we have a raw dataset to be processed, the preprocessing process in flask by uploading the crawled file and pressing the preprocessing button, the results will appear in the data frame. Then the dataset is downloaded and saved into excel in .csv format for further processing.

4.2.6. Word Weighting

Datasets that have gone through the preprocessing stage will then be labeled with sentences. Usually, labeling is done manually to get high and consistent accuracy [20]. However, in this study, labeling was carried out automatically using the texblob library. The weakness in using this texblob library is the inconsistent accuracy. This library is used because of its ease of use. Accuracy will be high when the tweets obtained have a sentence structure that is easily detected by the system. Another weakness is that when using the textblob library, tweets must be translated into English because it is not yet available for

sentences in Indonesian. Labeling will be done through preprocessed data which is processed into the Translator() function. After the dataset is translated into English, the data will be assigned a polarity using the sentiment method. polarity() consisting of more than zero (> 0) is positive, equal to zero ($= 0$) is neutral and less than zero (< 0) is negative. In addition, the text will be converted to numbers to be able to proceed to the classification stage. This conversion is carried out using the TF-IDF (Term Frequency– Inverse Document Frequency) method [17].

5. K-Nearest Neighbor Algorithm classification

The classification of the K-Nearest Neighbor algorithm in this analysis uses a ratio of 8:2, which is 80% training data and 20% testing data. There are 1409 data with queries available to see the results of accuracy and evaluation using a confusion matrix. In this process, the system can also upload datasets into the system for the classification process.
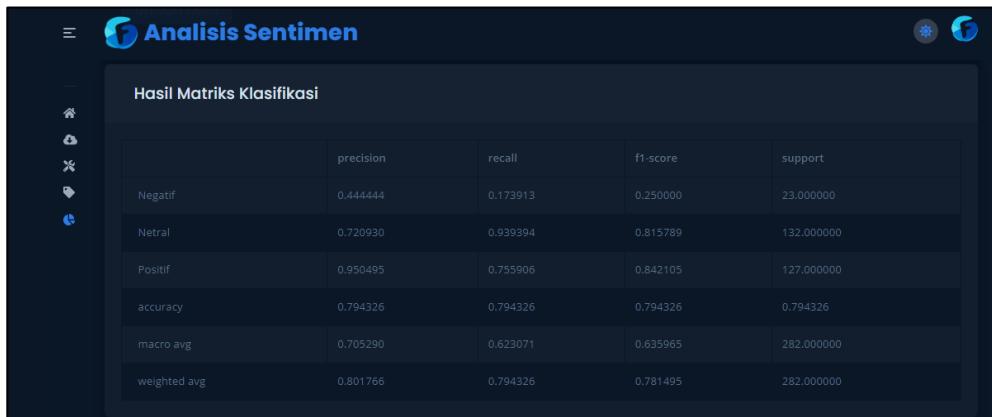
The classification results in an evaluation of the confusion matrix in the form of K-Nearest Neighbor accuracy, recall, f1-score, and precision. From the measurement results using 1409 data that have been taken, a classification trial was carried out. The results of the classification of the K-Nearest Neighbor algorithm can be seen in table 4.1:

Table 4.1 K-Nearest Neighbor Algorithm Classification

K	*Accuracy*	*Precision*	*Recall*	*f1-Score*
1	0.780142	0.914286	0.755906	0.827586
2	0.776596	0.881818	0.763780	0.818565
3	0.787234	0.914286	0.755906	0.827586
4	0.781350	0.821245	0.733567	0.800111
5	0.701090	0.813818	0.708680	0.821445
6	0.724467	0.848738	0.795276	0.821138
7	0.783688	0.914286	0.755906	0.827586
8	0.732470	0.921007	0.786531	0.849865
9	0.726532	0.958787	0.766214	0.819862
10	0.774356	0.932145	0.754325	0.821563
11	0.718387	0.946723	0.751210	0.808765
12	0.787234	0.923077	0.755906	0.831169
13	0.787234	0.931373	0.748031	0.829694
14	0.776596	0.913462	0.748031	0.822511
15	**0.794326**	**0.950495**	**0.755906**	**0.842105**
16	0.784671	0.950495	0.755906	0.842105
17	0.790780	0.915094	0.763780	0.832618
18	0.780142	0.914286	0.755906	0.827586

K	Accuracy	Precision	Recall	f1-Score
19	0.743546	0.848739	0.795276	0.821138
20	0.749857	0.853448	0.779528	0.814815

From the results of testing the value of K can be seen in Table 4.1, this study uses k=1, k=2, k=3, k=4, k=5, k=6, k=7, k=8, k=9, k=10, k=11, k=12, k=13, k=14, k=15, k=16, k=17, k=18, k=19, k=20. It can be concluded that K with the best classification results at k=15 with an accuracy of 79.43% testing the classification of the K-Nearest Neighbor algorithm can be seen in Figure 4.1:


Fig. 4.1 K-Nearest Neighbor Algorithm Classification Test

6. Evaluate with Confusion Matrix

Evaluation with the Confusion matrix of 1409 data using equations 2.1, 2.2, 2.3, 2.4. The parameters used are accuracy, precision, recall, and fi-score [18].

a. Accuracy is the predicted value from the calculation of true positive and true negative from the data taken. The calculation result of accuracy is 79.43. The translation of the calculation of accuracy is as follows:

$$\text{Accuracy} = \frac{(96 + 124 + 4)}{(96 + 31 + 0 + 3 + 124 + 5 + 2 + 17 + 4)} \text{ x } 100\%$$

$$\text{Accuracy} = \frac{224}{282} \text{ x } 100\%$$

Accuracy = 79.43%

b. Precision is the result of the calculation of the positive predicted true value divided by the total positive predicted value. The calculation result of precision is positive 0.95, neutral 0.72, and negative 0.44. The translation of the calculation of precision is as follows:

$$\text{Precision Positive} = \frac{96}{(96 + 3 + 2)} = \frac{96}{101}$$

Precision Positive = 0.95

$$\text{Precision Neutral} = \frac{124}{(31 + 124 + 17)} = \frac{124}{172}$$

Precision Neutral = 0.72

$$\text{Precision Negative} = \frac{4}{(0 + 5 + 4)} = \frac{4}{9}$$

Precision Negative = 0.44

c. Recall is the result of the calculation of the predicted true positive value divided by

the total positive value. The calculation result of the recall is positive 0.76, neutral 0.94, and negative 0.17. The description of the calculation of the recall is as follows:

Recall positive= $\frac{96}{(96 + 31 + 0)} = \frac{96}{127}$

Recall positive= 0,76

Recall neutral= $\frac{124}{(3 + 124 + 5)} = \frac{124}{132}$

Recall neutral= 0.94

Recall negative= $\frac{4}{(2 + 17 + 4)} = \frac{4}{23}$

Recall negative= 0.17

d. F1-score is the average of the results of precision and recall calculations. The calculation result of recall is positive 0.84, neutral 0.81, and negative 0.25. The elaboration of the calculation of the f1-score is as follows:

f1- Score Positive = 2 x $\frac{0,95 * 0,76}{0,95 + 0,76}$ =2 x $\frac{0,72}{1,71}$

f1- Score Positive = 0.84

f1- Score Netral= 2 x $\frac{0,72 * 0,94}{0,72 + 0,94}$ =2 x $\frac{0,67}{1,66}$

f1- Score Netral = 0.81
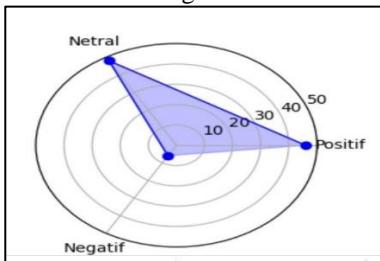
f1-Score Negative = 2 x $\frac{0,44 * 0,17}{0,44 + 0,17}$ =2 x $\frac{0,074}{0,61}$

f1- Score Negative = 0.25
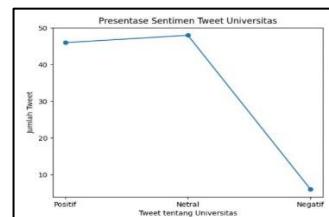
The most influential thing in the classification process is when the tweets that are obtained are still many tweets that use abbreviations, it will greatly affect the predicted value obtained and also affect the level of accuracy [19].

7. Visualisasi

The visualization is displayed as a percentage of the classification results, in the form of a bar chart, pie chart, and world cloud [21]. The researcher aims to display visualizations to make it easier to visually see the performance of the method used. Visualization results can be seen in Figure 4.8:
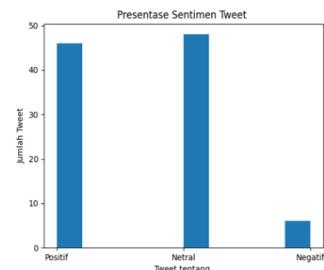
| (a) | (b) |

Fig 4.2. (a) WEB Form Visualization Interface.　　(b) Line Visualization Interface

(a)   (b)

Fig 4.3. (a) Cloud Visualization Interface   (b) Bar Chart Visualization Interface

8. Conclusion

The conclusions from the results of sentiment analysis research on universities using the K-Nearest Neighbor method using data as many as 1.409 tweets using university keywords are as follows: This research begins with crawling data, then the raw data is processed first in the preprocessing process. After that, the clean data is labeled (word weighting), then enters the classification stage using the K-Nearest Neighbor method with Euclidean distance calculations. The final stage is the evaluation stage of the method used with the confusion matrix with 80:20 split data, training data, and testing data. In this study, labeling was carried out automatically using the textblob library. The data used for labeling is data that has been preprocessed, then processed into the Translator() function. After the dataset is translated into English the data will be given a polarity sentiment. Polarity () consisting of more than zero ($>$0) is positive, equal to zero ($=$0) is neutral, and less than zero ($<$0) is negative. Based on the results of the study using 1.409 data using the K-Nearest Neighbor method, it obtained an accuracy of 79.43% at k=15, with positive sentiments of 47.59%, negative 8.66%, and neutral 43.75%. Quality of university services after a sentiment analysis of all university services can be concluded that university services for all facilities and infrastructure need to be improved so that in the future they can continue to develop and innovate.

Acknowledgements

References

1. L.K. Singh, D.R. Renuga. Student feedback sentiment analysis: A review. Mater Today Proc [Internet]. 2021;(xxxx). Available from: https://doi.org/10.1016/j.matpr.2020.10.782
2. M. Enkhsaikhan, W. Liu, E.J. Holden, P. Duuring. Auto-labelling entities in low-resource text: a geological case study. Knowl Inf Syst. 2021;63(3):695–715.
3. S. Bahassine, A. Madani, M. Al-Sarem, M. Kissi. Feature selection using an improved Chi-square for Arabic text classification. J King Saud Univ - Comput Inf Sci [Internet]. 2020;32(2):225–31. Available from: https://doi.org/10.1016/j.jksuci.2018.05.010

4. M.M. Agüero-Torales, J.I. Abreu Salas, A.G. López-Herrera. Deep learning and multilingual sentiment analysis on social media data: An overview. Appl Soft Comput. 2021;107.

5. W. Hachicha, L. Ghorbel, R. Champagnat, C.A. Zayani, I. Amous. Using process mining for learning resource recommendation: A Moodle case study. Procedia Comput Sci. 2021;192:853–62.

6. L. Chen, Y. Fu, S. You, H. Liu. Hybrid supervised instance segmentation by learning label noise suppression. Neurocomputing. 2022;496:131–46.

7. Wang R, Kwong S, Wang X, Jia Y. Active k -labelsets ensemble for multi-label classification. 2021;109.

8. Mohamad Beigi O, Moattar MH. Automatic construction of domain-specific sentiment lexicon for unsupervised domain adaptation and sentiment classification. Knowledge-Based Syst [Internet]. 2021;213(xxxx):106423. Available from: https://doi.org/10.1016/j.knosys.2020.106423

9. Zhao D, Gao Q, Lu Y, Sun D. Learning view-specific labels and label-feature dependence maximization for multi-view multi-label classification. 2022;124.

10. Dong X, Gu S, Zhuge W, Luo T, Hou C. Active label distribution learning. Neurocomputing. 2021;436:12–21.

11. Da'u A, Salim N, Rabiu I, Osman A. Weighted aspect-based opinion mining using deep learning for recommender system. Expert Syst Appl. 2020;140.

12. Jacinto R, Reis E, Ferrão J. Indicators for the assessment of social resilience in flood-affected communities – A text mining-based methodology. Sci Total Environ [Internet]. 2020;744:140973. Available from: https://doi.org/10.1016/j.scitotenv.2020.140973

13. Süzen N, Gorban AN, Levesley J, Mirkes EM. Automatic short answer grading and feedback using text mining methods. Procedia Comput Sci [Internet]. 2020;169(2019):726–43. Available from: https://doi.org/10.1016/j.procs.2020.02.171

14. Laksana EA, Suryana A, Heryono H. Evaluation of E-learning Activity Effectiveness in Higher Education Through Sentiment Analysis by Using Naïve Bayes Classifier. Sisforma. 2018;5(1):22.

15. Liklikwatil RD, Noersasongko E, Supriyanto C. Optimasi K-Nearest Neighbor Dengan Particle Swarm Optimization Untuk Memprediksi Harga Komoditi Karet. e-Jurnal JUSITI (Jurnal Sist Inf dan Teknol Informasi). 2018;7–2(2):172–82.

16. Ma Y, Huang R, Yan M, Li G, Wang T. Attention-based Local Mean K-Nearest Centroid Neighbor Classifier. Expert Syst Appl. 2022;201(April).

17. Mesarcik M, Ranguelova E, Boonstra A-J, van Nieuwpoort R V. Improving novelty detection using the reconstructions of nearest neighbours. Array. 2022;14(May):100182.

18. Mehbodniya A, Rao MV, David LG, Joe Nige KG, Vennam P. Online product sentiment analysis using random evolutionary whale optimization algorithm and deep belief network. Pattern Recognit Lett. 2022;159:1–8.

19. Min X, Qian K, Zhang B, Song G, Min F. Knowledge-Based Systems Multi-label active learning through serial – parallel neural networks. 2022;251.

20. Lai CH, Liu DR, Lien KS. A hybrid of XGBoost and aspect-based review mining with attention neural network for user preference prediction. Int J Mach Learn Cybern [Internet]. 2021;12(5):1203–17. Available from: https://doi.org/10.1007/s13042-020-01229-w

21. Chen X, Xie H. A Structural Topic Modeling-Based Bibliometric Study of Sentiment Analysis Literature. Cognit Comput. 2020;12(6):1097–129.